VOLUME 21 NO. 5 SEPTEMBER/OCTOBER 2000

BUILDING ENERGY SIMULATION

FOR USERS OF ENERGYPLUS, SPARK, DOE-2, BLAST, GENOPT, BUILDING DESIGN ADVISOR, ENERGY-10 AND THEIR DERIVATIVES

User News

What's New ?

VisualSPARK 1.0 Release ....

VisualSPARK 1.0 is now available! See the article on p. 10. To purchase VisualSPARK, go to

http://SimulationResearch.lbl.gov

EnergyPlus Beta 4 ....

The fourth planned beta test version of EnergyPlus was released in October. To get a no-cost license for Beta 4 go to

http://www.gard.com/eplustest.htm

If you already have a license for testing previous versions of EnergyPlus, you don't need a new license for Beta 4.

DOE-2 In the News ....

The July 2000 issue of Canada's *Advanced Building Newsletter (ABN)* featured an article on the 4 Times Square Building, claimed to be the largest "green" speculative office tower in the world. DOE-2 was used extensively in the design process. Email Nils Larsson to receive the *Advanced Building Newsletter* (larsson@ @greenbuilding.ca). You can also read a short article about 4 Times Square on the Steven Winter Associates web site at www.swinter.com/ projects/ 4times_sq.html

What's Inside ?

Note: The DOE-2 Directory of Software and Services, List of Resource Centers and List of Consultants will appear in the next issue of the User News.

The Bild-IT Project: Acquiring Geometry from CAD for EnergyPlus Through the Use of Industry Foundation Classes

by
Rob Hitchcock
Lawrence Berkeley National Laboratory

The Commercial Building Systems Group (CBSG) at Lawrence Berkeley National Laboratory (LBNL) is involved in an international joint development project[1] to create a computer-based tool called Bild-IT [2] that will support integrated building design. Bild-IT will implement interoperability among building services with the initial effort focused on transferring the geometry of a building design, drawn with commercial CAD software, to the EnergyPlus building simulation program. This geometry data transfer is based on the International Alliance for Interoperability's (IA) Industry Foundation Classes (IFC) data model and uses the BSPro COM-Server middleware (see "The BSPro COM-Server: Interoperability Among Software Tools Using Industry Foundation Classes" on p. 4).

The International Alliance for Interoperability (IAI) and Industry Foundation Classes (IFC)

The International Alliance for Interoperability (IAI)[3] was formed in 1994 with the mission of defining, publishing, and promoting specifications for a common data model for the AEC/FM[4] industry that would support interoperability among the wide variety of existing and future software tools used by industry participants. The promise of interoperability is that it would facilitate the sharing of building information worldwide, throughout the building project life cycle, and across all disciplines and technical applications.

This common data model, called the Industry Foundation Classes (IFC), is an object-oriented specification of the attributes of, and relationships between, building related entities such as walls and windows (*tangible objects*), and spaces and performance metrics (*intangible objects*). Several versions of the evolving IFC model have been released by the IAI beginning with Release 1.0 in 1996. Commercial software capable of importing and exporting IFC data is now available. The first software tools on the market are primarily CAD, and are compliant with IFC Release 1.5.1, the first IFC version with official compliance testing defined by the IAI. IFC Release 2x was officially released by the IAI in London in late October 2000. The documentation and specifications for Release 2x are available for download from the United Kingdom Chapter of the IAI[5] and the LBNL CBSG web site[6].

Creating IFC-Based Software

In the past, the process of creating software capable of reading and writing IFC data files was a difficult and costly one. Developers have had to invest considerable time and effort either developing this capability from scratch or purchasing and learning to use one of the few available (and costly) development toolboxes. A wider range of options is now becoming available as more developers take the plunge into the IFC world.

Over the past year, LBNL has been developing IFC-based software using alpha and beta versions of the BSPro COM-Server middleware by Olof Granlund Oy. This middleware provides a library of methods (functions) for extracting data from (and writing data to) IFC files. The server is targeted at IFC data that are relevant to building services software tools. The recently available commercial version of this middleware tool is primarily targeted at geometry. A key advantage of using this middleware is that it is presently compatible with IFC Release 1.5.1 and Release 2.0, and has officially passed certification testing

---

[1] The other partners are the Canadian branch of **AEA Technology**, a UK-based CFD developer, the **Halton Group**, a Finnish manufacturer of HVAC equipment; and **Olof Granlund Oy**, a Finnish building services consulting firm. Bild-IT work in the U.S. is supported by the U.S. Department of Energy, program manager Drury Crawley.

[2] *Bild-IT: An Integrated Design Tool for the HVAC Industry*, http://eetd.lbl.gov/news/NL3/bild.it.html

[3] http://iaiweb.lbl.gov/

[4] Architects, Engineers, Contractors/Facility Managers

[5] http://cig.bre.co.uk/iai_uk/

[6] http://eetd.lbl.gov/btp/iai/index.html

for R1.5.1. Future versions of the server will support all extant releases of the IFC model.

CAD to EnergyPlus

A beta version of a client to the BSPro COM-Server has been developed for acquiring geometry from an IFC file created by an IFC-compliant CAD tool and transferring this geometry to EnergyPlus. This client was developed as a Windows DLL using Visual C++ 6.0. The current version of the client can acquire geometry from IFC R1.5.1 files and R2.0 files.

As shown in Fig. 1, the EnergyPlus BSPro client reads an existing IFC data file, extracts instances of relevant IFC objects from the file, maps these objects to EnergyPlus Input Data Dictionary (IDD) objects, and creates an EnergyPlus Input Data File (IDF). This extraction and mapping process includes these IFC objects: IfcProject, IfcSite, IfcBuilding, IfcBuildingStorey, IfcSpace, IfcWall, IfcWindow, IfcDoor, and IfcSlab. Only the geometry of these object instances is currently passed to the IDF. Additional IDD object data, such as construction material characteristics and simulation parameters, are defaulted and written to the IDF so that a preliminary EnergyPlus run can be executed.

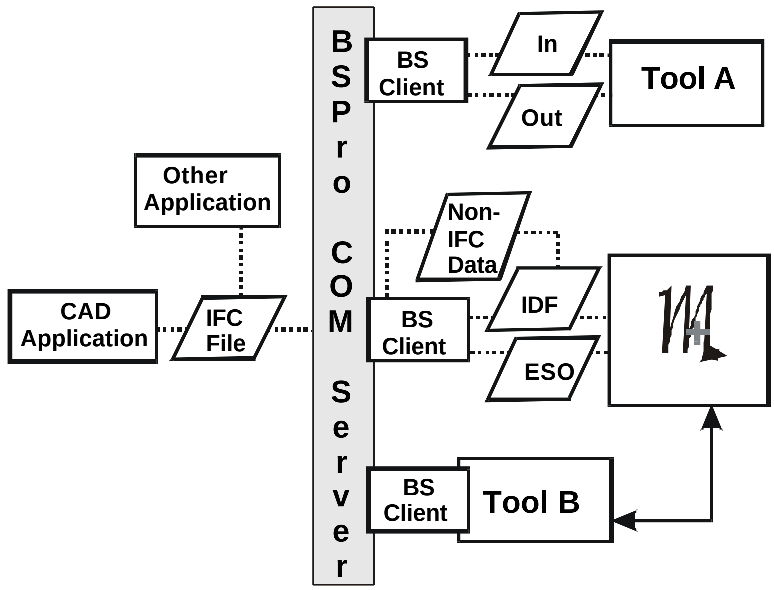


Figure 1: Acquiring geometry from CAD via a BSPro COM-Server client.

The principal output of this run is a DXF file that allows a visual review of the input geometry using standard CAD software. The client-generated IDF can be input directly to EnergyPlus or can be manually edited (e.g., using IDFEditor). Manual editing can be used to modify the default characteristics written to the IDF, or to add missing elements like space gains and HVAC system descriptions.

At the present time, each instance of IfcSpace is mapped to a thermal zone in EnergyPlus. This means that someone using an IFC-compliant CAD tool as a front end for energy simulation should create IfcSpace instances from a thermal perspective, rather than from an architectural room perspective. In the future it is expected that either CAD or customized user interfaces will take advantage of the existing IfcZone class definition, which aggregates spaces and partial spaces into a proper thermal zone. This is a user interface issue rather than an IFC data model issue.

As the IFC model is extended to include new objects relevant to energy simulation, and the available IFC-compliant software tools provide a user interface for defining these building data, both BSPro COM-Server and the EnergyPlus client will be enhanced. It is planned that runtime copies of the BSPro COM-Server and the EnergyPlus client will be made available to registered users of the EnergyPlus Version 1.0 release.

LBNL contacts for the BILD-IT project are Dr. Philip Haves (phaves@lbl.gov) and Dr. Rob Hitchcock (rjhitchcock@lbl.gov).

The BSPro COM-Server:
Interoperability Among Software Tools Using Industry Foundation Classes

by
Antti Karola and Hannu Lahtela
OLOF GRANLUND OY, CONSULTING ENGINEERS
PO Box 59 (Malminkaari 21)
FIN-00701 Helsinki
FINLAND

antti.karola@granlund.fi

Tel. +358 9 3510 31
Fax +358 9 3510 3424
www.granlund.fi

The IFC standard is here — it is time to develop compliant software

The major constraint on the everyday use of powerful calculation tools at different stages of the building design process has been time-consuming manual data input, especially building geometry data. However, the continuing development of the IFC standard[1] has created completely new possibilities to achieve interoperability for design software and has taken real advantage of the data repository and open data transfer offered by the IFC object model of the building.

Most of the latest architectural CAD tools from Autodesk, Graphisoft and Nemetschek are already IFC compliant[2]. In particular, the current version of the IFC includes building geometry information needed for thermal simulation programs. To take real advantage of this feature, it is time for software developers to make their software IFC compliant.

A new tool to develop IFC-compliant software without detailed knowledge of the standard

Nevertheless, when applying the IFCs, our experience has been that the software developer needs wide knowledge of the standard, because of the complexity of the IFC object model. It is difficult to read the huge amount of building design data stored in an IFC file and extract the information needed by a particular application. To make this work easier for developers not familiar with the IFCs, Olof Granlund Oy (Finland) has developed a new middleware tool, the BSPro COM-Server for IFC Files. By using this tool, software developers can easily achieve IFC compatibility.

Software interoperability by BSPro COM-Server

Based on Microsoft's COM technology, BSPro COM-Server is a so-called *middleware* solution that can link new or existing software tools to allow an exchange of IFC-compliant building data. It uses a language-independent architecture and can be used in any programming environment within Windows™.

The handling of IFC files using BSPro COM-Server is possible without a deep understanding of the IFC standard. The software developer only needs to install the BSPro COM-Server and then write program code into his/her software to create a BSPro Client Module. The methods offered in the BSPro COM interface enable easy access to IFC classes and their properties.

Where to use BSPro COM-Server?

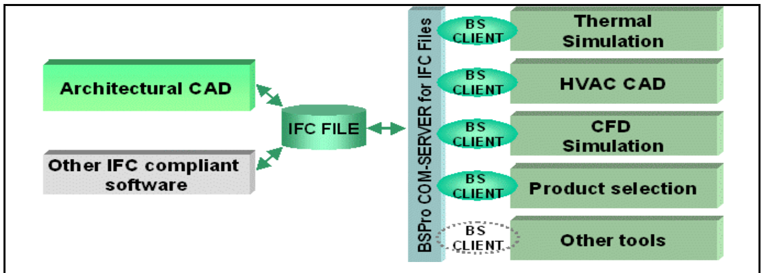
BSPro COM-Server has its scope in Building Services ("BS"), i.e., HVAC and electrical. Currently, another program, called BSPro, handles building geometry and thermal data for the building envelope. BSPro COM-Server can be used with different tools throughout the building services implementation process (design, manufacturing, contracting and facilities management).

---

[1] The IFC standard is published and promoted by International Alliance for Interoperability (IAI), see http://iaiweb.lbl.gov/ .
[2] IFC version 1.5.1

When using the IFC standard, the work of manual data input diminishes because the needed data can be imported directly from the IFC file created by the architect. This has great benefits, especially when using thermal and CFD simulation programs.

The simulation tools can be integrated with the IFC model. This enables, for instance, dimensioning and comparisons of design alternatives in the early stages of the design process. The easier handling of IFC files enables updating and adding information into them during the whole design process.



Software interoperability by BSPro COM-Server for IFC Files

In the future, the IFC model of the building, also containing data from thermal simulations, gives many possibilities for data reuse in other IFC compliant design programs, such as HVAC-CAD. For example, the sizing of terminal boxes and ventilation system ductwork can be based on simulation data imported from the IFC model.

Olof Granlund Oy is currently using BSPro COM-Server in RIUSKA, thermal simulation software based on DOE-2.1E. The building geometry stored in the IFC file is imported into the simulation tool using BSPro Client routines added into RIUSKA. After performing simulations, thermal data can be exported from RIUSKA to the same IFC file thus updating the information within that file.

BSPro COM-Server has also been beta tested during its development process by several software vendors in collaboration with Olof Granlund Oy. The following testers have been using it in their software development projects:

- AEA Technology: CFX software
- Massachusetts Institute of Technology: MIT-CFD software.
- Lawrence Berkeley National Laboratory: EnergyPlus energy software (see article on p. 2)

BS Pro COM-Server available in November

BSPro COM-Server will be available on the market in November 2000. The tool can be downloaded from http://www.granlund.fi. Currently it is IFC 1.5.1 compliant, but in the future newer versions of the IFC model will also be supported.

How to contact us:

Simulation Research Group

MS: 90-3147

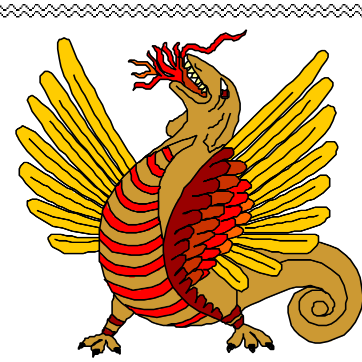
LAWRENCE BERKELEY NATIONAL LABORATORY

Berkeley, CA 94720-0001 U.S.A.

Fax: (510) 486-4089

Email: KLEllington@lbl.gov

Web: http://SimulationResearch.lbl.gov



Run for safety, foolish pedestrians!

These reports are available from the Simulation Research Group's web site at http://SimulationResearch.lbl.gov. Click on "The Latest News" or "Reports" under Publications.

Recent LBNL Reports

LBNL-46303

A Bottom-Up Engineering Estimate of the Aggregate Heating and Cooling Loads of the Entire U.S. Building Stock

Yu Joe Huang
Lawrence Berkeley National Laboratory

Jim Brodrick
D&R International

Abstract
A recently completed project for the U.S. Department of Energy's (DOE) Office of Building Equipment combined DOE-2 results for a large set of prototypical commercial and residential buildings with data from the Energy Information Administration (EIA) residential and commercial energy consumption surveys (RECS, CBECS) to estimate the total heating and cooling loads in U.S. buildings attributable to different shell components such as windows, roofs, walls, etc., internal processes, and space-conditioning systems. This information is useful for estimating the national conservation potentials for DOE's research and market transformation activities in building energy efficiency. The prototypical building descriptions and DOE-2 input files were developed from 1986 to 1992 to provide benchmark hourly building loads; they include 112 single-family, 66 multi-family, and 481 commercial building prototypes. The study consisted of two distinct tasks: 1) perform DOE-2 simulations for the prototypical buildings and develop methods to extract the heating and cooling loads attributable to the different building components; and 2) estimate the number of buildings or floor area represented by each prototypical building based on EIA survey information. These building stock data were then multiplied by the simulated component loads to derive aggregated totals by region, vintage, and building type. The heating and cooling energy consumption of the national building stock estimated by this bottom-up engineering approach was found to agree reasonably well with estimates from other sources, although significant differences were found for certain end-uses. The main added value from this study, however, is the insight it provides about the contributing factors behind this energy consumption, and what energy savings can be expected from efficiency improvements for different building components by region, vintage, and building type.

LBNL-46304

Improving DOE-2's RESYS Routine: User-Defined Functions to Provide More Accurate Part-Load Energy Use and Humidity Predictions

Hugh I. Henderson, Jr.,
CDH Energy Corp.

Danny Parker
Florida Solar Energy Center

Yu Joe Huang
Lawrence Berkeley National Laboratory

Abstract
In hourly energy simulations, it is important to properly predict the performance of air conditioning systems over a range of full and part load operating conditions. An important component of these calculations is to properly consider the performance of the cycling air conditioner and how it interacts with the building. This paper presents improved approaches to properly account for the part load performance of residential and light commercial air conditioning systems in DOE-2. First, more accurate correlations were given in order to predict the degrada-tion of system efficiency at part-load

These reports are available from the Simulation Research Group's web site at http://SimulationResearch.lbl.gov. Click on "The Latest News" or "Reports" under Publications.

Recent LBNL Reports

conditions. In addition, a user-defined function for RESYS was developed that provided improved predictions of air conditioner sensible and latent capacity at part-load conditions. The user function also provided more accurate predictions of space humidity by adding *lumped* moisture capacitance into the calculations. The improved cooling coil model and the addition of moisture capacitance predicted humidity swings that were more representative of the performance observed in real buildings.

LBNL-46006

Enhancing and Extending the Capabilities of the Building Heat Balance Simulation Technique For Use in EnergyPlus

Richard Strand
School of Architecture
Univ. of Illinois at Urbana-Champaign, IL

Fred Winkelmann, Fred Buhl, Joe Huang
Lawrence Berkeley National Laboratory
Berkeley, CA, USA

Richard Liesen, Curtis Pedersen, Daniel Fisher, Russell Taylor
Building Simulation Laboratory
University of Illinois at Urbana-Champaign, IL

Drury Crawley
U.S. Department of Energy
Washington, DC

Linda Lawrie
U.S. Army CERL
Champaign, Illinois

Abstract

With the advent of the computing age, heat-balance based techniques for simulating thermal loads in buildings became a reality for architects and engineers. However, since the 1970s, the capabilities of most of the well-known heat-balance based simulation programs have remained fairly stagnant. Much of the reason behind this trend lies with the complexity of the programming required to deal with the fundamental physics encountered in a building and the relative simplicity of the programming languages that were available. With the ever-increasing capabilities of the desktop personal computers and the improved features of the modern programming languages, it is now possible and prudent to revisit the basic heat-balance formulation and investigate how its capabilities can be expanded. This paper discusses some of the technical details behind recent advances in heat-balance based simulation capabilities achieved by the team of researchers developing the EnergyPlus program for the U. S. Department of Energy. The EnergyPlus project seeks to combine the best features of the DOE-2 program and the IBLAST program (research version of BLAST). This paper focuses on the marriage of the basic heat-balance engine of BLAST with advanced simulation ideas from the IBLAST and DOE-2 programs. This complex task requires careful attention to algorithmic integrity as well as overall program construction and data management. This paper provides the theoretical background for several of the enhancements to the heat-balance based simulation technique used in EnergyPlus.

These reports are available from the Simulation Research Group's web site at http://SimulationResearch.lbl.gov. Click on "The Latest News" or "Reports" under Publications.

Recent LBNL Reports

LBNL-46302

A Joint US-China Demonstration Energy Efficient Office Building

Mary Beth Zimmerman
U.S. Department of Energy

Yu Joe Huang
Lawrence Berkeley National Laboratory

Rob Watson
Natural Resources Defense Council

Shi Han
Administrative Centre for China's Agenda 21

Ron Judkoff and Micah Sherman
National Renewable Energy Laboratory

Abstract
In July 1998, USDOE and China's Ministry of Science and Technology (MOST) signed a Statement of Work to develop a demonstration energy-efficient office building and demonstration center in Beijing that will eventually house the Administrative Center for China's Agenda 21 (ACCA21). The statement calls for

- the Chinese side to be responsible for the basic construction of the 13,000 m^2 9-story building,
- the US side for technical assistance and the incremental costs of the energy efficiency improvements, and
- the joint establishment of a Demonstration Center to provide outreach and exhibit energy-efficient building technologies.

The US technical team made several trips to China to meet with ACCA21 and the design team, and used the DOE-2.1E simulation program to analyze the energy performance of a preliminary building design and study alternative designs and energy-efficient strategies. A feasibility study completed in September found the largest and most cost-effective savings potentials in reducing cooling and lighting energy use, and identified eight generic measures in lighting, windows, daylighting, and HVAC systems and controls. Following these and other recommendations from the US team, the design team produced a schematic cross-shaped building design that, based on the DOE-2 analysis, lowered total energy use by 40% compared to standard practice. While the design and analysis were underway, a task force called ACCORD21 (American-Chinese Council Organized for Responsible Development in the 21st Century) was formed in April 1999 under the leadership of NRDC to solicit support and contributions from U.S. industry, A/E firms, and universities. Two design workshops were held, first in Pittsburgh and then in Beijing, that brought together the Chinese and US project participants and produced further refinements and energy-efficiency improvements to the building design. As of June 2000, the authors are completing the final energy analysis and selection of energy-efficiency measures. Construction is expected to begin in the early part of 2001.

Building Design Advisor 2.0

Decision making through the integrated use of multiple simulation tools and databases

The **Building Design Advisor** (**BDA)** is a Windows program that addresses the needs of building decision-makers from the initial, schematic phases of building design through the detailed specification of building components and systems. The BDA is built around an object-oriented representation of the building and its context, which is mapped onto the corresponding representations of multiple tools and databases. It then acts as a ***data manager*** and ***process controller***, automatically preparing input to simulation tools and integrating their output in ways that support multi-criterion decision making. The latest public release of BDA (version 2.0) is linked to three main applications:

- A **Schematic Graphic Editor (SGE)**, for graphic input of building components and systems,
- **DElight**, a simplified daylighting simulation tool, and
- the **DOE-2.1E** building energy simulation program.

The following **enhancements** have been made to BDA 2.0 (as of 09/15/00):

- Greater flexibility in project development with features such as "Save as.."
- Greater user control over object properties with editing of Solution and Story properties, building azimuth, etc.
- User interface enhancements allow easier navigation of the building model with less ambiguities.
- Several bug fixes.
- Extended documentation.

Current research and development efforts are focused on the development of links to:

- **Desktop Radiance**, a Windows 95/98/NT version of the **Radiance** lighting/daylighting simulation and rendering software, and
- **Athena**, a life-cycle analysis of embodied energy and environmental impact of materials.

The minimum and recommended system **requirements** to run the BDA software are as follows:

Minimum	**Recommended**
Pentium 75	Pentium 200 or better.
Windows 95, 98, NT 4.0.	Windows 95, 98, NT 4.0.
16 / 32MB RAM under Windows 95	24 / 64MB RAM under Windows NT 4.0.
30 MB of larger hard disk space.	60 MB of larger hard disk space.
640x480 or higher screen resolution.	1024x768 or higher screen resolution.

The BDA source code is available for licensing; if interested, please contact Dr. Papamichael at K_Papamichael@lbl.gov.

To learn more about the BDA software and to download a copy of the latest public version, please visit

http://kmp.lbl.gov/BDA

VisualSPARK

VisualSPARK allows you to build customized models of complex physical processes by connecting calculation objects. It is aimed at the simulation of innovative and/or complex building systems.

Release of Version 1.0

VisualSPARK 1.0 has been released. Its main elements are a **user interface**, a **network specification language**, a **solver** for solving simultaneous algebraic and differential equations, and a **results processor**. With the network specification language you link the calculation objects into networks that represent a building's envelope and/or HVAC systems. The solver solves this network for user-specified input parameters. With the results processor you graphically display the results of the calculation.

The UNIX version of VisualSPARK runs under the SunOS, Solaris, Linux and HPUNIX operating systems. The PC version of VisualSPARK runs under Windows 95/98/NT/2000.

VisualSPARK Features:

- solves non-linear systems of arbitrary complexity
- solves systems with a few equations to thousands of equations
- user-specified time step
- robust solution methods
- easy to change variables from input to calculated
- HVAC component library
- dynamic plotting: plot results while simulation is running
- 10 to 20 times faster execution times than similar programs in many cases (due to automatic problem partitioning and reduction in number of iteration variables)

An end-user license for VisualSPARK costs $250. To purchase the program, go to http://SimulationResearch.lbl.gov.

If you would like to get an idea of what the program does before purchasing it, you can review the SPARK User's Manual, which can be downloaded from http://SimulationResearch.lbl.gov > SPARK > SPARK User's Manual.

VisualSPARK was developed by the LBNL Simulation Research Group and Ayres Sowell Associates, with support from the U.S. Department of Energy.

http://SimulationResearch.lbl.gov > SPARK

THERM 2.1

Two-Dimensional Building Heat Transfer Modeling

THERM is windows-based software for modeling heat-transfer effects in building components such as windows, walls, foundations, roofs, doors, appliances, and other products where thermal bridges are of concern. The program's two-dimensional heat-transfer analysis, based on the finite-element method, allows you to evaluate a product's energy efficiency and local temperature patterns, which may relate directly to problems with condensation, moisture damage, and structural integrity.

THERM is available free of charge from the Building Technology Department at Lawrence Berkeley National Laboratory. Download from

http://windows.lbl.gov/software/therm/therm_getacopy.htm

ENERGY-10, Version 1.3

Version 1.3 of ENERGY-10 is now available. It includes the much-anticipated ***WeatherMaker*** function. *WeatherMaker* allows users to create their own weather files based on information available from nearly 4,000 weather stations throughout the U.S. Revisions to the program itself include some minor fixes, an improved and expanded Help section, and greater clarity in titling and identification of various sections. Contact the Sustainable Buildings Industries Council for more information, or to order your upgrade disc (the cost is $15, which covers production and shipping).

ENERGY-10, written in C^{++}, is a design tool for smaller residential or commercial buildings that are less than 10,000 ft^2 floor area, or buildings that can be treated as one- or two-zone increments. It performs whole-building energy analysis for 8760 hours/year, including dynamic thermal and daylighting calculations. ENERGY-10 was specifically designed to facilitate the evaluation of energy-efficient building features in the very early stages of the design process.

Input: Only four inputs required to generate two initial generic building descriptions. Virtually everything is defaulted but modifiable. As the design evolves, the user adjusts descriptions using fill-in menus (utility-rate schedules, construction details, materials).

Output: Summary table and 20 graphical outputs available, generally comparing current design with base case. Detailed tabular results also available.

Platform: PC-compatible, Windows 3.1/95/98, Pentium processor with 16 MB of RAM is recommended.

Sustainable Buildings Industries Council

Douglas K. Schroeder
Associate Director
1331 H Street, NW, suite 1000
Washington, D.C. 20004 USA

Tel: (202) 628-7400 ext 210
Fax: (202) 393-5043
SBICouncil@sbicouncil.org
www.psic.org/energy10.htm

The Building Energy Simulation User News is published bi-monthly and distributed electronically by the Simulation Research Group at Lawrence Berkeley National Laboratory, with cooperation from the Building Systems Laboratory at the University of Illinois. Direct comments or submissions to Kathy Ellington, MS: 90-3147, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, or email KLEllington@lbl.gov or fax us at (510) 486-4089. Direct BLAST-related inquiries to the Building Systems Laboratory, email support@blast.bso.uiuc.edu or phone (217) 333-3977 © 2000 Regents of the University of California, Lawrence Berkeley National Laboratory. This work was supported by the Assistant Secretary for Energy Efficiency and Renewable Energy, Office of Building Technology, State and Community Programs, Office of Building Systems of the U.S. Dept. of Energy, under Contract No. DE-AC03-76SF00098

GenOpt® 1.1: Beta 2 Version

The Beta 2 version of GenOpt 1.1 has been released. It contains an additional algorithm for multi-dimensional optimization, new algorithms for one-dimensional optimization, and an algorithm for parametric runs in a multi-dimensional space. The new version also allows processing of multiple function values and has an improved graphical user interface.

GenOpt is a multi-parameter optimization program, available free of charge from LBNL. It automatically finds the values of user-selected design parameters that minimize an *objective function*, such as annual energy use, calculated by an external simulation program like EnergyPlus, SPARK, DOE-2, BLAST, TRACE, TRNSYS, etc. GenOpt can be used with any simulation program that has text-based input and output. It also offers an interface for adding custom optimization algorithms to its library.

Genopt 1.1, Beta 2 (with user manual) may be downloaded from

http://SimulationResearch.lbl.gov > GenOpt

Honey is Sweet, but the Bee Stings

DOE-2 DOE-2 DOE-2 DOE-2 DOE-2

PC Version of DOE-2.1E from ESTSC

DOE-2.1E (version 107) for Windows is available from the Energy Science and Technology Software Center (ESTSC). Previously, ESTSC licensed only UNIX and VAX versions. This updated version of DOE-2 incorporates bug fixes and new features such as a Cooled Beam HVAC system and polygon input for walls, floors and ceilings. Like previous DOE-2.1E products from ESTSC, this version accepts textual BDL input but does not have a graphical user interface. Cost of DOE-2.1E-WIN (Version 107) is:

$ 300 U.S. Government, non-profit Educational
$ 575 U.S., Mexico, Canada
$ 1075 Other Foreign

To order, call Ed Kidd or Walt Kelly at ESTSC (865) 576-2606, or email to estsc@adonis.osti.gov.

DOE-2.1E Documentation Update

Corrections to Appendix A (Hourly Report Variables) of the DOE-2.1E *Supplement* may be downloaded from the SRG web site (http://SimulationResearch. lbl.gov). Click on "Documentation" under DOE-2 in the left menu. You want "Update Package #3."

DOE-2 Help Desk

Bruce Birdsall - Phone/Fax: (925) 671-6942, M-F 10 a.m. to 3 p.m. PDT.
Contact Bruce if you have a DOE-2 problem or question. If you need to send a fax, please be sure to phone him first. This is a free service, supported by the U. S. Department of Energy.

DOE-2 Training

DOE-2 courses for beginning and advanced users: phone Marlin Addison at (602) 968-2040, or send email to marlin.addison@doe2.com

DOE-2 DOE-2 DOE-2 DOE-2 DOE-2

DOE-2.1E Bugs 96 through 111

Following are bug fixes 96 through 111. Shown at the left is the version number of DOE-2.1E, which is incremented for each new set of fixes. There follows a short description of the set of fixes, the initials of the author and date of fix. Note that a particular version will include all fixes made up to and including that version number. So, Version -111 includes all prior fixes. You can determine which version of DOE-2.1E you are using by checking any of the output reports, where version *NNN* is indicated as "DOE-2.1E-*NNN*". The complete DOE-2.1E bug list may be accessed from our website at http://SimulationResearch.lbl.gov

-096 : sys wth

For SYSTEM-TYPE=PIU, when TERMINAL-TYPE=SERIES-PIU and INDUCED-AIR-ZONE is set equal to the zone itself, the System simulation aborts in the DESIGN routine. There is no work around aside from avoiding the particular input that triggers the problem. [WFB 12/17/98]

wth: Create a format for measured data. (ident name is MSWTH)
wth: Add CD144S decoder. (ident name is CD144S)

-097 : bdl dedt lds plt sys

Fix various problems in the inside surface temperature calculation: -- Remove overwrite of input function name by variable <FREM> in BDL subroutine WFMAIN. The modification ident name is MODST1 .

- Fix misspelled variables in SURFTL. Caused problem with inside surface temperature calculation when doors were present. [EE 990202]
- For consistency, make length of /LOCALD/ common block the same in Loads, Systems and Plant (no effect on calculation results).
- Add inside surface temperature calculation flag to /LOCALD/.

(no effect on calculation results). [WFB 11/15/99]
The modifications ident name is MODST1 .

-098 : bdl drlc lds sim sys

Fix benign errors that some compilers complain about. Fix two problems with PARAMETRIC-INPUT (parametric runs):

(1) The parametric run feature depended on being able to open a single physical file multiple times with different unit numbers. This is called a "shared file" or "file sharing." The Unix F77 compiler allows this by default, and Digital (now COMPAQ) Visual Fortran allows it if the file is opened with the correct OPEN statement parameters set (SHARED and/or SHARE). But some compilers may not allow file sharing. This mod eliminates the need for file sharing in a parametric run. This change should have no impact on the user.

(2) When an input with an include file (##include) was followed by PARAMETRIC-INPUT's, BDL would give fatal errors (ILLEGAL COMMAND IN PARAMETRIC RUN). This mod fixes the problem - include files can now be used in conjunction with parametric runs. [WFB 5/13/99]

Apply metric/English conversion to frame area and frame U-value.

Apply metric/English conversion to frame area and frame U-value in verification report LV-H. Previously the conversion was not applied, so that these values were printed in English units even in metric output runs. [FCW 09/14/99]

Fix some variables in Loads that are used without EDTT names. [EE 990202]

Use the same outside air film resistance in LV-C, LV-D and LV-H. Previously, LV-D used RO = 0.4 and LV-C and LV-H used RO = 1./3.3476 = 0.2987. We will now use RO = 0.1957 in all of these reports. This corresponds to a 15 mph wind speed (ASHRAE winter conditions). Add a missing comdeck for PIU (fixes bug in -094). Fix bug in RESYS system where, in some cases, program was getting a bad value for VENT-TEMP-SCH when venting was enabled and VENT-TEMP-SCH was not input. [EE 11/16/99]

-099 : lds

For daylighting availability calculation, add Perez method of calculating luminous efficacy from measured solar when present on the weather file (e.g., TMY and TMY2). The Perez method does not require atmospheric turbidity values (ATM-TURBIDITY keyword). For weather files without measured solar (e.g., TRY) the previously-used CIE method for calculating luminous efficacy is retained. As before, the CIE method requires atmospheric turbidity values to be input. Both the Perez and CIE methods require atmospheric precipitable moisture values. These are now calculated hourly from dewpoint temperature; the ATM-MOISTURE keyword has been removed. Because the Perez method is based on measured correlations it is expected to give more accurate luminous efficacy values than the CIE method, which relies on atmospheric turbidity values that are usually poorly known. [FW 4/15/99]

-100 : bdl dedt dkey lds

Implement 2-D and 3-D polygons for EXTERIOR-WALL, ROOF, INTERIOR-WALL, UNDERGROUND-FLOOR and UNDERGROUND-WALL. Previously these surfaces had to be rectangles. Add loads verification report LV-N to print geometry in building coordinates. [EE 6/25/98]

-101 : bdl edt rlc sim wth

Increase EDTT table size. Change base source code from VAX-VMS to a version compatible with Sun Fortran 77 and Digital Visual Fortran 6.0. This includes converting file open statements, fdate call and DUMPIT routine. The base source code now compiles on both Sun Fortran 77 and DVF 6.0 without modification. [EE 11/24/99]

-102 : bdl dedt dkey drlc sys

Implement COOLBEAM system type. This model for this system was a collaborative effort between LBNL and Olof Granlund Co.(Helsinki). The ident name in the source code is clbeam. [FB 991124]

-103 : bdl

Replace the cycling AC EIR-FPLR curves with Hugh Henderson's "good" curve, which corresponds to a degradation coefficient of .05 . The previous curves had higher degradation at part load. [FB 12/15/99]

-104 : dkey sys

Remake global tables for input functions to account for mods 100 and 102. [EE 11/23/99]

-105 : lds sys

Changes some variables that had the form AA(pointer+...) to EDTT variable form. This had no effect on the calculations. [EE 12/15/99]

-106 : bdl

In Input Functions, give error message for expressions like x=1.23(y+z) that are missing an operator between 1.23 and (y+z). [EE 12/15/99]

-107 : bdl sim

Print version number and fatal error count in log file. [EE 12/20/99]

-108 : bdl sim

If there are errors, return error status number (=104 for doebdl, =105 for doesim) so that the command shell can give proper diagnostic message. [EE 2/10/2000]

-109 : sys

In the COOLBEAM system, for very high humidity loads and resultant high inlet water temperature, the zone temperature calculation was failing to converge. The result could be seen in the hourly reports as an hour-to-hour temperature fluctuation, as well as a fluctuation in beam output. [FB 4/12/2000]

-110 : drlc

In the cooling tower hourly report, FAN ELEC and PUMP ELEC should have units of kW not Btu. [FB 4/12/2000]

-111 : sys

Fix bug in surface temp HOURLY-REPORT problem. Systems was getting incorrect results when there is more than one zone in a system and SURF-TEMP-CALC=YES in BUILDING-LOCATION. Speed up surface temperature calculations. (related to IDENT MODST) [EE 2000.10.03]

Software Available From Lawrence Berkeley National Laboratory

Downloads	
BDA 2.0 (Building Design Advisor)	kmp.lbl.gov/BDA
COMIS (multi-zone air flow and contaminant transport model)	www-epb.lbl.gov/comis
EnergyPlus™ (new-generation whole-building energy analysis program, combining best features of BLAST and DOE-2.	To beta test EnergyPlus go to SimulationResearch.lbl.gov > EnergyPlus
GenOpt® **1.1** (generic optimization program)	SimulationResearch.lbl.gov > GenOpt
RADIANCE (analysis and visualization of lighting in design)	radsite.lbl.gov/radiance/
Desktop Radiance (integrates the Radiance Synthetic Imaging System with AutoCAD Release 14)	radsite.lbl.gov/deskrad/
RESEM (Retrofit Energy Savings Estimation Model) (calculates long-term energy savings directly from actual utility data)	eetd.lbl.gov/btp/resem.htm
SUPERLITE (calculate illuminance distribution for room geometries)	eetd.lbl.gov/btp/superlite20.html
THERM 2.1 (model two-dimensional heat-transfer effects in building components where thermal bridges are of concern)	windows.lbl.gov/software/therm/therm.html
WINDOW 4.1 (thermal analysis of window products)	windows.lbl.gov/software/window/window.html

Request by Fax from 510.486.4089	
RESFEN 3.1 (choose energy-efficient, cost-effective windows for a given residential application)	windows.lbl.gov/software/resfen/resfen.html

Web Based	
Home Energy Saver (quickly compute home energy use)	hes.lbl.gov

Purchase	
SPARK (Simulation Problem Analysis and Research Kernel) (build simulations of innovative building envelope and HVAC systems by connecting component models)	For Windows, SUN, Linux, go to SimulationResearch.lbl.gov > SPARK
ADELINE 2.0 (day/lighting performance in complex spaces)	radsite.lbl.gov/adeline/

Meetings, Conferences, Symposia

FEMP
Energy 2001 Workshop

To be held
June 4-6,2001 in Kansas City, MO
Contact: Rick Klimkos (FEMP)
Tel: 202.586.8287
fax: 202.586.3000
Net: http://www.energy2001.ee.doe.gov

CLIMA 2001

To be held
September 15-18, 2001 in Naples, Italy
Contact the secretariat at
Tel: +39.02.55.193.446
Email: clima@clima2000.it
Net: http://www.clima2000.it

9th National Conference on Building Commissioning

To be held
May 9-11, 2001, in Cherry Hills, NJ
Contact: Carolyn Dasher, Conference Manager
Tel: 503.248.4636 x 204

Fax: 503.295.0820
Email: cdasher@peci.org
Net: http://www.peci.org/ncbc

I B P S A

BUILDING SIMULATION 2001

To be held
August 13-15, 2001 in Rio de Janeiro, Brazil
All information may be found at the BS2001 web site:
WWW.LABEEE.UFSC.BR/BS2001/

ASHRAE Winter Meeting
To be held
January 27-31 in Atlanta, GA

ASHRAE Annual Meeting
To be held
June 23-27, 2001 in Cincinnati, OH

Contact: jyoung@ashrae.org
ASHRAE Meetings Section
1791 Tullie Circle NE
Atlanta, GA 30329
Tel: 404.636.8400 -- Fax: 321.5478
Net: http://www.ashrae.org

Love All … Trust a Few

[redacted]news

www.bso.uiuc.edu

Building Systems Laboratory (BSL)
30 Mechanical Engineering Building
University of Illinois
1206 West Green Street
Urbana, IL 61801
Telephone: (217) 333-3977
Fax: (217) 244-6534
support@blast.bso.uiuc.edu

The **Building Loads Analysis and System Thermodynamics (BLAST** program predicts energy consumption, energy system performance and cost for new or existing (pre-retrofit) buildings.

BLAST contains three major sub-programs:

- **Space Load Prediction** computes hourly space loads in a building based on weather data and user inputs detailing the building construction and operation.
- **Air Distribution System Simulation** uses the computed space loads, weather data, and user inputs.
- **Central Plant Simulation** computes monthly and annual fuel and electrical power consumption.

Heat Balance Loads Calculator (HBLC)
The BLAST graphical interface (HBLC) is a Windows-based interactive program for producing BLAST input files. You can download a demo version of HBLC (for MS Windows) from the BLAST web site (User manual included).

HBLC/BLAST Training Courses
Experience with the HBLC and the BLAST family of programs has shown that new users can benefit from a session of structured training with the software. The Building Systems Laboratory offers such training courses on an as needed basis typically at our offices in Urbana, Illinois.

WINLCCID 98
LCCID (Life Cycle Cost in Design) was developed to perform Life Cycle Cost Analyses (LCCA) for the Department of Defense and their contractors.

To order BLAST-related products, contact the Building Systems Laboratory at the address above.

Program Name	Order Number	Price
PC BLAST Includes: BLAST, HBLC, BTEXT, WIFE, CHILLER, Report Writer, Report Writer File Generator, Comfort Report program, Weather File Reporting Program, Control Profile Macros for Lotus or Symphony, and the Design Week Program. The package is on a single CD-ROM and includes soft copies of the BLAST Manual, 65 technical articles and theses related to BLAST, nearly 400 processed weather files with a browsing engine, and complete source code for BLAST, HBLC, etc. Requires an IBM PC 486/Pentium II or compatible running MS Windows 95/98/NT.	3B486E3-0898	$1500
PC BLAST Package Upgrade from level 295+	4B486E3-0898	$450
WINLCCID 98: executable version for 386/486/Pentium	3LCC3-0898	$295
WINLCCID 98: update from WINLCCID 97	4LCC3-0898	$195

The last four digits of the catalog number indicate the month and year the item was released or published. This will enable you to see if you have the most recent version. All software will be shipped on 3.5" high density floppy disks unless noted otherwise.

Design Optimization with GenOpt®

Michael Wetter
Simulation Research Group
MWetter@lbl.gov

Abstract

Although simulation is increasingly being used in building system design, the full potential of simulation is usually not achieved. To improve system performance, designers generally guess different values of system parameters and redo the simulation. This is inefficient and labor intensive. Also, if the number of parameters being varied exceeds two or three, the designer can be overwhelmed in trying to understand the nonlinear interactions of the parameters. However, techniques exist that allow automatic, multidimensional optimization of a simulation model, leading to better design with less effort.

We describe how such optimization can be done using GenOpt, a generic optimization program. We show an example of how to use GenOpt to design an office building such that source energy consumption for heating, cooling, and lighting is minimal with respect to selected design parameters.

In this example, the optimization yields 14% energy savings. The additional time required to set up the optimization is about an hour. The measures found by using optimization not only decrease operating costs, but also lead to better daylighting usage, which results in higher comfort for the building occupants.

1 Introduction

Setting up an input file for a thermal building simulation is time intensive. Once the analyst has set up the input files, she/he does not utilize the potential that computer simulation offers in optimizing the design. Today, "optimizing" a building or HVAC system consists of guessing parameter settings that may lead to better system performance. This is very time consuming. Also, if the number of parameters being varied exceeds two or three, the analyst has difficulty understanding the interaction of the system parameters in order to make an educated guess that leads to further improvement. Hence, only limited improvement is achieved.

However, mathematical optimization allows determining the values of system parameters that lead to optimal system performance using a computational procedure. LBNL's Simulation Research Group developed GenOpt, a generic optimization program, that allows such optimization. Doing the optimization requires only little additional labor compared to preparing the simulation input. GenOpt can be used with any simulation program that has text based I/O, such as EnergyPlus, DOE-2, SPARK[1], BLAST [2], TRNSYS[3], etc. or any user-written code.

[1]For more information on EnergyPlus, DOE-2, and SPARK, see `http://SimulationResearch.lbl.gov`
[2]For more information on BLAST, see `http://www.bso.uiuc.edu`
[3]For more information on TRNSYS, see `http://sel.me.wisc.edu/TRNSYS`

Our goal here is to show step by step how an optimization with GenOpt 1.1 can be done on a typical office building, and how optimization improves building design. The simulations will be done with EnergyPlus 1.0β4, and we will minimize annual source energy consumption for heating, cooling, and lighting.

We will first introduce some terminology, and explain the building being optimized, before we outline the procedure that allows us to perform the optimization. In the last section, we will present the results of the optimization.

2 Terminology

By solving a minimization problem, a set of **independent parameters** (also called **design parameters** or **free parameters**) will be modified to find a minimum of a function. We will denote this set by $x \in \mathbb{R}^n$. The independent parameters are typically values such as insulation thickness, window sizes, volume flow rate, etc. Clearly, they cannot take on any value, since, e.g., a window area cannot be negative. Therefore, the independent parameters are constrained to a **feasible set**. In building simulation, most parameters can be constrained by specifying a lower and upper bound. These kinds of constraints are called **box constraints**. The function being minimized is called the **objective function**. It typically stands for a value such as annual energy consumption, operating cost, peak load, etc.

The set of independent parameters that minimizes the objective function is called **minimizer** (or minimum point). A minimizer may be a **local minimizer** or a **global minimizer**. It is generally impossible to know whether a local minimizer is also a global minimizer. However, finding a local minimizer is already an improvement in system design compared to the initial values of x.

To solve an optimization problem, an optimization algorithm is initialized with some parameter set that is in the feasible set. The algorithm then selects iteratively new parameter sets such that a minimizer is found without having to evaluate the objective function on the whole feasible set. This is usually done by checking the behavior of the function locally around a point, and by applying some strategy – based on the local function shape and previous iterations – which hopefully leads to a large step towards a minimizer.

3 Optimization Problem

We will optimize the yearly source energy consumption for heating, cooling, and lighting of an office building in Chicago. It is assumed that the energy consumption of the thermal zone shaded in Figure 1 is representative for the overall building energy consumption for heating, cooling, and lighting. In Figure 1, floor, ceiling, and north and south wall are adiabatic. The exterior walls have a U-value of $0.25\,\mathrm{W/(m^2\,K)}$ and consist of (listed from outside to inside) 1 cm wood siding, 10 cm insulation and 20 cm concrete. The ceiling and floor consist of carpet, 5 cm concrete, insulation and 18 cm concrete. Interior walls are 12 cm brick. Both windows are low-emissivity, double pane with Krypton gas fill and exterior shading device. The shading device is actuated only during summer when the solar radiation on the window exceeds $200\,\mathrm{W/m^2}$. Both windows have a fixed shading overhang of 1 m depth. The zone's daylighting control dims the light to an illumination setpoint of 500 lux at a distance of 3 m from each window. The heating set point is 20°C between 6 AM and 11 PM and 18°C otherwise. The cooling set point is a constant 25°C.

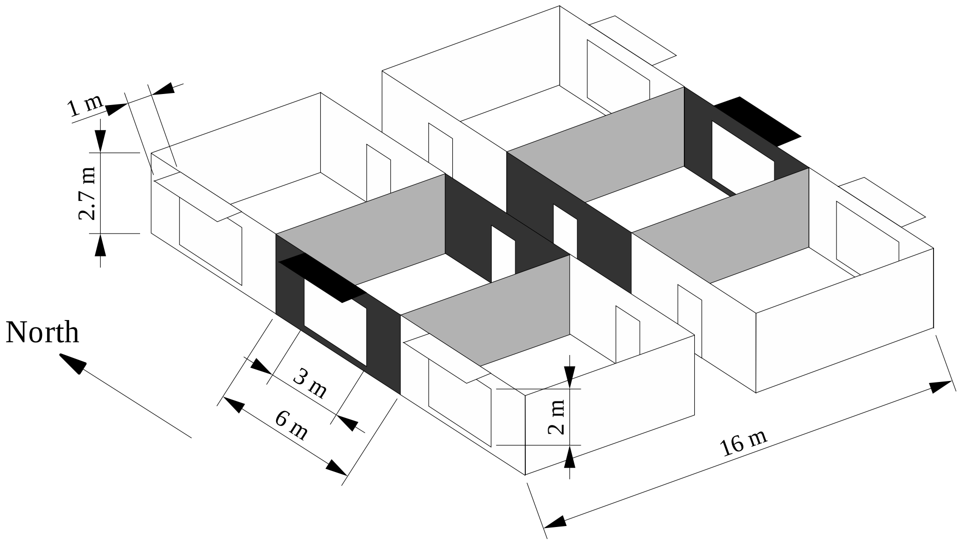


Figure 1: Optimized office zone in its initial configuration

The yearly source energy consumption is calculated by

$$E_{tot}(x) = \frac{Q_{heat}(x)}{\eta_{heat}} + \frac{Q_{cool}(x)}{\eta_{cool}} + 3\, E_{lights}(x), \tag{1}$$

where Q denotes the zone's yearly heating or cooling load, E_{lights} the zone's electricity consumption for lighting, and the efficiencies $\eta_{heat} = 0.44$ and $\eta_{cool} = 0.77$ are typical plant efficiencies that relate the zone load to the source energy consumption required for heating and cooling generation, including electricity consumption for fans and pumps. Lighting (as well as electricity for fans and pumps) is weighted by a factor 3 to convert electricity to fuel energy usage. η_{heat} and η_{cool} are obtained from [HF99] and are typical of large office buildings in Chicago.

The independent parameters are the building azimuth, α, the width of the west window, w_w, the width of the east window, w_e, and the transmittance of the movable exterior shading device, τ (assumed to be the same for solar and visible radiation). Table 1 shows the lower bounds, l, the initial values, x_0, and the upper bounds, u, of the independent parameters. The bounds on the window width have been chosen such that the window is not wider than the wall. Since the objective function is periodic in the building azimuth, we did not constrain α.

How much progress can be achieved by using optimization relative to the initial values, x_0, certainly depends on how good the initial conditions already are. For our test case, we selected x_0 such that we believed our building design is optimal with respect to $E_{tot}(x)$. The window area has been selected such that electricity consumption for lighting can be kept small due to large daylighting use. We believe that further increasing the window size would lead to higher cooling loads due to increased solar gains, without increasing daylighting consumption significantly. Also, since heating energy consumption is usually not a dominant factor for office buildings, increased solar gains in winter would not reduce the total energy consumption significantly. Further, it turned out that rotating the building by 90° while keeping the other parameter fixed decreases $E_{tot}(x)$ only by 2%. Therefore, we left the initial condition for the azimuth at 0°, as in the first guess of the design.

	l	x_0	u
building azimuth α [DEG]	$-\infty$	0	$+\infty$
width west window w_w [m]	0.1	3	5.9
width east window w_e [m]	0.1	3	5.9
shading device transmittance τ	0.2	0.5	0.8

Table 1: Lower bound, l, initial value, x_0, and upper bound, u, of the four independent parameters

As in all building simulations, equation (1) is discontinuous with respect to the independent parameter, x. Furthermore, for some values of x, the gradient of (1) with respect to x tends to infinity. Hence, we cannot use a gradient based optimization algorithm. We will therefore use a so-called pattern search algorithm developed by Hooke and Jeeves [HJ61] which does not use the gradient of (1).

4 Setting up the Optimization Problem

We assume that the simulation input files have already been written. We are now faced with the task of (i) specifying what parameters of the input files have to be varied within which range, (ii) what number in the output file has to be minimized, (iii) what optimization algorithm has to be used, (iv) how the simulation program is to be started, and (v) where the input files and output files are located.

While we outline all necessary steps to set up an optimization run, we have to refer to the GenOpt manual for a more detailed explanation of the various keywords, as well as for the mathematical description of the optimization algorithm. Based on Figure 2, we will first outline how GenOpt performs the simulation runs: GenOpt reads the **simulation input template files**, which are nothing but the simulation input files where the numerical values of each independent parameter was replaced by its variable name. GenOpt then replaces these variable names by numerical values (which were determined by the chosen optimization algorithm) and writes the **simulation input files**. Afterwards, GenOpt starts the simulation program, waits until the simulation is completed, and then reads the **simulation log files** and the **simulation output files**. The simulation log files indicate whether the simulation had an error. If not, GenOpt searches the objective function value in the simulation output files. An **initialization file** tells GenOpt where all files are located. The **command file** lists the independent parameters and their bounds. It also specifies what algorithm should be selected from GenOpt's algorithm library to perform the optimization. The **configuration file** tells GenOpt what messages in the log file indicate a simulation error, and how GenOpt has to start the optimization.

4.1 Specification of Independent Parameters

To specify what values of the simulation input files are independent parameters, we need to replace each numerical value that has to be optimized with a variable name. To distinguish variable names from other text in the input files, each variable name needs to be enclosed in percentage signs. To do so, we make a copy of the **simulation input file**, which we will call **simulation input template file**. Then, if a section of the simulation input file looks like

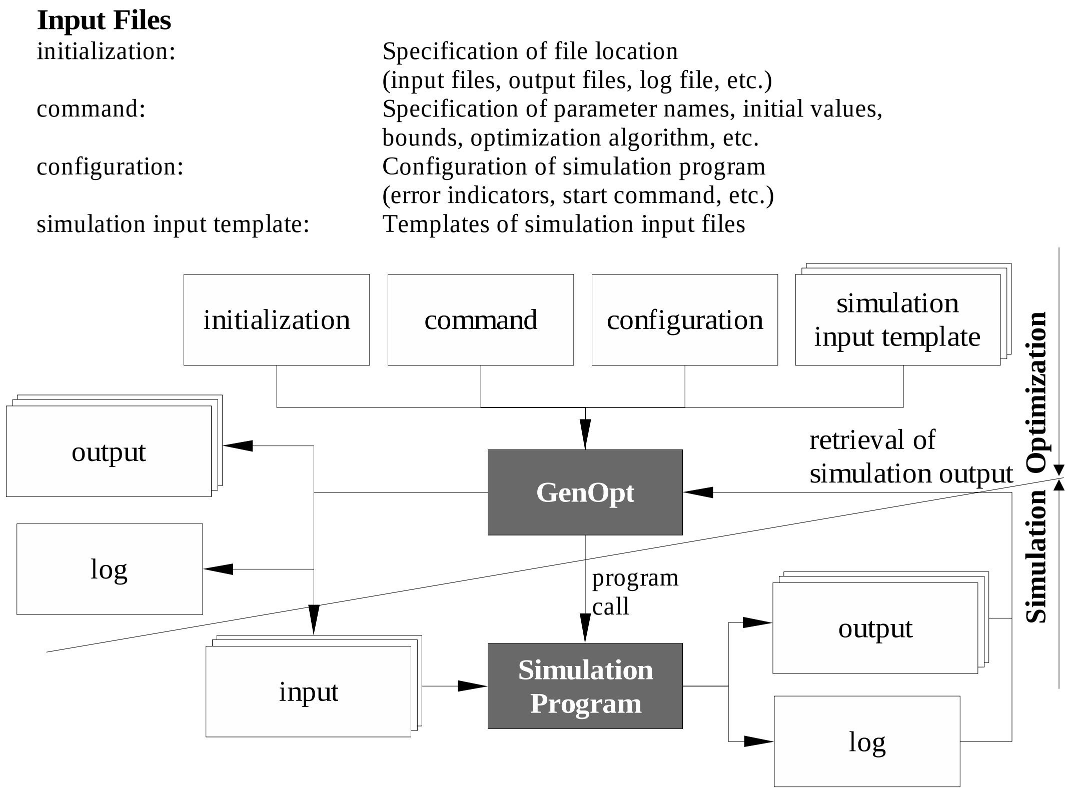


Figure 2: Interface between GenOpt and the simulation program that calculates the objective function

```
    office zone,   ! Building Name
             90,   ! Building Azimuth
              1,   ! Building Terrain
```

and we want to vary the building azimuth, we need to modify the section according to

```
    office zone,   ! Building Name
      %azimuth%,   ! Building Azimuth
              1,   ! Building Terrain
```

`azimuth` will now be the name of one of the independent parameters. The same name has to be specified in the **optimization command file**. The specification of all independent parameters listed in Table 1 looks like

```
Parameter{ Name = azimuth;  Min = SMALL; Ini = 0;   Max = BIG; Step =  10; }
Parameter{ Name = w_we_win; Min = 0.1;   Ini = 3;   Max = 5.9; Step = 0.2; }
Parameter{ Name = w_ea_win; Min = 0.1;   Ini = 3;   Max = 5.9; Step = 0.2; }
Parameter{ Name = tau;      Min = 0.2;   Ini = 0.5; Max = 0.8; Step = 0.1; }
```

The `Parameter` section shown above specifies for each variable the minimum value, the initial value, and the maximum value. It also declares a step size, which is a value that is used by the optimization algorithm (see the GenOpt manual for more information on step size).

4.2 Specification of Algorithm

GenOpt has a library with different optimization algorithms. In the **optimization command file**, we can specify what algorithm has to be used. A typical section looks like

```
Algorithm{ Main = HookeJeeves;  StepReduction = 0.5;  NumberOfStepReduction = 2; }
```

This specification causes GenOpt to use the Hooke-Jeeves optimization algorithm, and to pass two additional parameters to the algorithm. (See the GenOpt manual for an explanation of the parameters.)

In the **optimization command file**, we also need to specify a section `OptimizationSettings` where we set the maximum number of iterations and some other parameters that are commonly used by all optimization algorithms. It typically looks like

```
OptimizationSettings{ MaxIte = 200; MaxEqualResults = 50; WriteStepNumber = false; }
```

4.3 Specification of Simulation Program

We also need to specify how GenOpt can start the simulation program, and what messages in the simulation log file indicate an error of the simulation program. This information is stored in the **simulation configuration file**. GenOpt's installation already contains such files for various simulation programs (e.g., DOE-2, SPARK, EnergyPlus, TRNSYS, etc.). The user does not need to modify these files.

4.4 Function Value to be Minimized

Since most simulation programs write their output in text files that contain the function values and additional text, we need to specify how GenOpt can find the objective function value in the simulation output file. Consider an EnergyPlus output file that looks like

```
...
5,1,Cumulative Days of Simulation[] ! When Run Period Report Variables Requested
100,2,ZONE ONE,Lights-Electric Consumption[J] !RunPeriod
120,2,ZONE1AIR,Purchased Air Heating Energy[J] !RunPeriod
122,2,ZONE1AIR,Purchased Air Cooling Energy[J] !RunPeriod
End of Data Dictionary
1,CHICAGO IL UNITED STATES TMY2 94846,  41.78, -87.75,  -6.00, 190.00
5,365
100,9.8254090E+09
120,2.4119670E+09
122,1.0562006E+10
End of Data
```

Then, the function values that we want to read from the file are beyond the strings `5,`, `120,`, `122,` and `100,`. Therefore, we will specify in the **optimization configuration file** a section of the form

```
ObjectiveFunctionLocation{
      Delimiter1 = "5,"  ;   Name1 = "E_tot";
      Delimiter2 = "120,";   Name2 = "Q_heat";
      Delimiter3 = "122,";   Name3 = "Q_cool";
      Delimiter4 = "100,";   Name4 = "E_lights";  }
```

Note that `E_tot` actually refers to the *cumulative days of simulation*. This is a work-around since EnergyPlus cannot compute equation (1). Hence, we will compute (1) directly in GenOpt by adding one line of code after reading the function values. To be able to report all values, namely $E_{tot}(x)$, $Q_{heat}(x)$, $Q_{cool}(x)$, and $E_{lights}(x)$, we need to read in *any* value for $E_{tot}(x)$ from the simulation output file. This will assign a variable in GenOpt for $E_{tot}(x)$ that we can then overwrite using equation (1). We stress that modifying GenOpt's code is usually not required, since most simulation programs allow the computation of (1) directly in the simulation program. Hence, with most programs, no code modification of GenOpt would be necessary. Rather than modifying GenOpt's source code, we could also have used some post-processing program to compute (1) or any econometrics calculation, e.g., to compute operating costs.

4.5 File Location

As the last step, we have to specify in the **optimization configuration file** where all the files in Figure 2 are located. After doing that, we can start the optimization.

5 Result of Optimization

To solve the optimization problem, we used the Hooke-Jeeves pattern search algorithm [HJ61], with improvements by Smith [Smi69], Bell and Pike [BP66], and De Vogelaere [DV68].

Using this algorithm, the total source energy consumption, $E_{tot}(x)$, was reduced by 14% compared to the initial values. Table 2 shows the initial values and the values corresponding to the minimizer of (1), and Figure 3 shows the values at each iteration step. To achieve the minimum point, 117 iterations were required. Since the algorithm required the objective function value at some points more than once (GenOpt detects such cases and, hence, does not perform a simulation run), a total of 98 EnergyPlus simulations were required. This took 4 hours on a 200MHz Pentium computer running Windows NT 4.

As shown in Table 2, the optimal building design has been achieved by increasing the window width and by rotating the building by 90° (the window initially facing west is now facing north). Even though the larger window width leads to slightly higher cooling energy consumption (due to increased solar gains), the electricity consumption for lighting was reduced by 19%. Note that this finding strongly depends on the fact that the building has an exterior shading device and a window overhang. Both measures reduce solar heat gains during summer. Also, due to increased solar gains, heating energy consumption has been reduced by 28%.

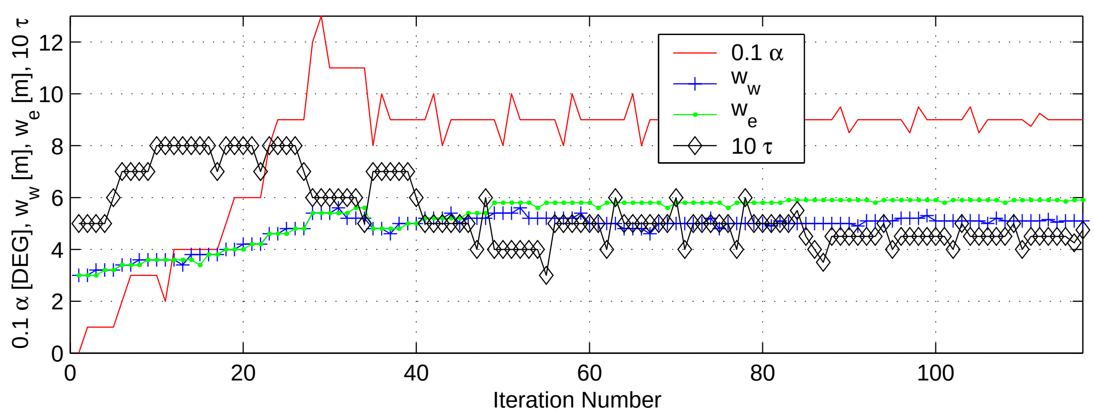
As can be seen from Figure 3, increasing τ in the 5th iteration reduced $E_{tot}(x)$ (otherwise, the algorithm would not have further increased τ). Hence, at the original building configuration, an increase in the shading device transmittance reduces $E_{tot}(x)$. This may be explained by the west-east window orientation, which causes large solar gains since, at sunrise and sunset, the window overhang is not very effective and the solar incidence angle is small. However, as the building is rotated by 90°, τ reduces to an optimal value of 0.45.

To show that the Hooke-Jeeves algorithm really converged to a minimizer, we also did parametric runs on 2400 possible parameter configurations, and an optimization using another algorithm[4]. It

[4]The simplex method of Nelder and Mead [NM65] with an extension of O'Neill [O'N71]

	initial value	optimum value
Source energy consumption E_{tot} [kWh/(m^2 a)]	168.1 100%	145.3 86%
Heating energy consumption Q_{heat} [kWh/(m^2 a)]	9.86 100%	7.14 72%
Cooling energy consumption Q_{cool} [kWh/(m^2 a)]	32.0 100%	34.5 108%
Lighting energy consumption E_{lights} [kWh/(m^2 a)]	34.7 100%	28.0 81%
building azimuth α [DEG]	0	90
width west window w_w [m]	3	5.1
width east window w_e [m]	3	5.9
shading device transmittance τ	0.5	0.45

Table 2: Initial and final values of optimization



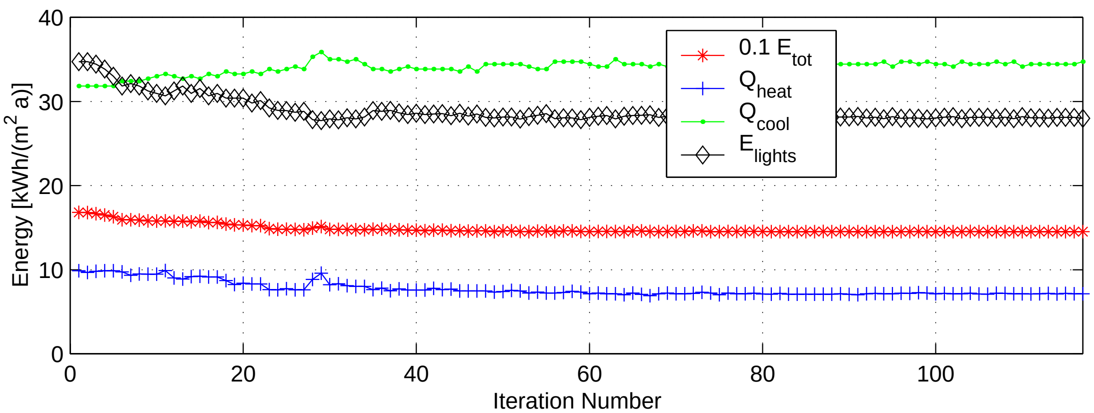


Figure 3: Iteration sequence of optimization algorithm

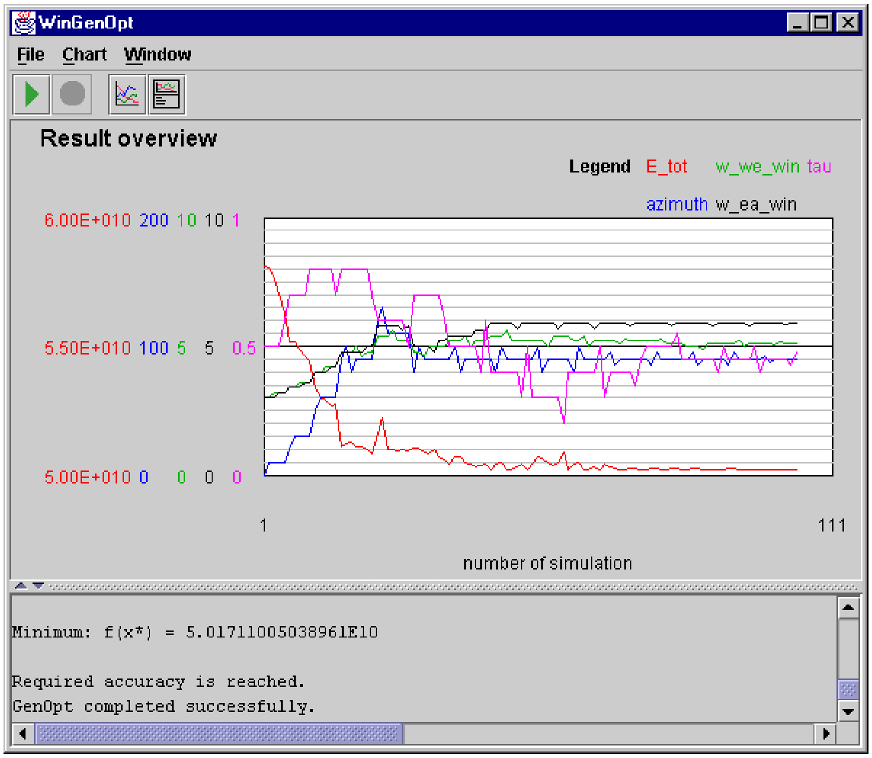


Figure 4: GenOpt's display of the progress of an optimization run

turned out that both optimization algorithms converged to the same point. Also, the parametric runs did not reveal any parameter combination with a lower function value. Therefore, the solution obtained with the optimization is very likely the global minimizer.

In summary, the optimization leads to 14% lower source energy consumption (and hence lower energy costs) and better daylighting usage.

6 Availability

GenOpt 1.1 is expected to be released in November 2000. Like GenOpt 1.0, it will be downloadable free of charge from `http:\\SimulationResearch.lbl.gov`. The user manual can also be downloaded from this web site. Since GenOpt is written entirely in Java 1.2, it can be run on any operating system that supports Java (e.g., Windows NT/95/98/2000, Unix, Linux). The installation also contains simulation configuration files for different building simulation programs, such as EnergyPlus, DOE-2, SPARK and TRNSYS.

GenOpt can either be run as a console application (for example, from a batch job or shell script to do multiple optimization runs), or with a graphical user interface that shows the progress of the optimization (Figure 4). GenOpt 1.1 has four different optimization algorithms, two for multidimensional optimization (Hooke-Jeeves algorithm [HJ61] and simplex method of Nelder and Mead

[NM65] with an extension of O'Neill [O'N71]) as well as two line-search methods (Golden Section and Fibonacci interval division). The line-search methods allow the optimization of a function with respect to one independent parameter in a unimodal interval.

7 Conclusion

Since GenOpt is a stand-alone program that is configured for the used simulation program by text files only, it can be used to do optimization with *any* simulation program that has text-based input and output. Also, since GenOpt processes whatever ASCII input file it is given, the system being optimized can be far more complex than the one shown here. Hence, it is also possible to optimize building envelope design together with HVAC system design as long as the independent parameters are continuous.

If the number of independent parameters exceeds two, the designer usually cannot understand and quantify the nonlinear interactions of the various system parameters. For such cases, mathematical optimization is a cost-effective tool that supports the analyst in designing better systems, which often results in lower operating costs and higher comfort for the building occupants. Doing optimization also gives the designer a better understanding of the system behavior. The time required for setting up an optimization problem is usually less than one hour.

8 Acknowledgements

This work was supported by the US Department of Energy (DOE), the Swiss Academy of Engineering Sciences (SATW), the Swiss National Energy Fund (NEFF), and the Swiss National Science Foundation (SNSF). The author would like to thank these institutions for their generous support.

References

[BP66] M. Bell and M.C. Pike. Remark on algorithm 178. *Comm. ACM*, 9:685–686, September 1966.

[DV68] R. De Vogelaere. Remark on algorithm 178. *Comm. ACM*, 11:498, Jul. 1968.

[HF99] Joe Huang and Ellen Franconi. Commercial heating and cooling loads component analysis. Technical Report LBL-37208, Lawrence Berkeley National Laboratory, EETD, November 1999.

[HJ61] R. Hooke and T.A. Jeeves. 'Direct search' solution of numerical and statistical problems. *J. Assoc. Comp. Mach.*, 8(2):212–229, April 1961.

[NM65] J.A. Nelder and R. Mead. A simplex method for function minimization. *The computer journal*, 7(4):308–313, January 1965.

[O'N71] R. O'Neill. Algorithm AS 47-function minimization using a simplex procedure. *Appl. Stat. 20*, 20:338–345, 1971.

[Smi69] Lyle B. Smith. Remark on algorithm 178. *Comm. ACM*, 12:638, November 1969.